Noname manuscript No.
(will be inserted by the editor)

Manifold regularization in structured output space for semi-supervised structured output prediction

Fei Jiang · Lili Jia · Xiaobao Sheng · Riley LeMieux

Received: date / Accepted: date

Abstract Structured output prediction aims to learn a predictor to predict a structured output from a input data vector. The structured outputs include vector, tree, sequence, etc. We usually assume that we have a training set of input-output pairs to train the predictor. However, in many real-world applications, it is difficult to obtain the output for a input, thus for many training input data points, the structured outputs are missing. In this paper, we discuss how to learn from a training set composed of some input-output pairs, and some input data points without outputs. This problem is called semi-supervised structured output prediction. We propose a novel method for this problem by constructing a nearest neighbor graph from the input space to present the manifold structure, and using it to regularize the structured output space directly. We define a slack structured output for each training data point, and proposed to predict it by learning a structured output predictor. The learning of both slack structured outputs and the predictor are unified within one single minimization problem. In this problem, we propose to minimize the structured loss between the slack structured outputs of neighboring data points, and the prediction error measured by the structured loss. The

F. Jiang
College of Fine Arts, Shanghai University, Shanghai 200444, China

X. Sheng
School of Economics and Management, Tongji University, Shanghai 200092, China
E-mail: sxbao765@hotmail.com
Corresponding author

L. Jia, X. Sheng
The 3rd Research Institute of Ministry of Public Security, Shanghai 200031, China

R. LeMieux
Department of Computing and Information Sciences, Kansas State University, Manhattan, KS 66506, United States
E-mail: riley.lemieux@hotmail.com

problem is optimized by an iterative algorithm. Experiment results over three benchmark data sets show its advantage.

Keywords Structured output prediction · Structured loss · Manifold regurlarization · Neighborhood smoothness · Gradient descent

1 Introduction

1.1 Background

In machine learning community, the problems of pattern classification and regression has been studied well. Classification and regression are two most popular supervised learning problems [33,18,4,24,39,17,25,31,37,36,28,29,30,27, 26]. In these problems, we usually have a training set of input-output pairs. The task is to train a predictive model from the training set to predict the output of a test input. In both the problems of classification and regression, the input is usually a feature vector. The output of classification problems is a binary class label, which represents a positive class or a negative class. The output of regression problems is a continues response variable. Recently, it is proposed that the output of a machine learning problem can be beyond a binary label and a continues response, and the output is structured in many real-world applications [3,10,20,13,12,32,14]. For example, in multi-class classification problems, the output is a vector presenting which class the input belongs to. In hierarchical classification problems, the classes are organized as a tree, and each class is a node of the tree. Moreover, in natural language parsing problems, the output of a input language sequence is a sequence. When the structured output is considered, the transitional predictive model learning algorithms cannot be used because the output does not to them. To solve this problem, the structured output prediction problem is proposed to learn a specific given structured output. This problem assume a training set of input-structured output pairs are available for the learning problem. However, in real-world applications, it is usually expensive or time-consuming to obtain a structured output for a input data point. Thus in many cases, we have a limited number of input-structure output pairs, and a large number of inputs without corresponding structured outputs. In this case, we try to learn a predictive model with a large number of input data points and a small number of structured outputs. This problem is call semi-supervised structured output prediction [5,21,15]. In this paper, we investigate this problem, and proposed a novel method to solve it.

1.2 Related works

There are some existing works on semi-supervised structured output prediction problem. We introduce them as follows.

- Altun et al. [2] proposed the problem of predicting multiple inter-dependent outputs by learning in a semi-supervise setting, and a method to solve this problem. The method is a maximum-margin method, and it uses the manifold of the input data space by exploring both the labeled and unlabeled data points. Moreover, this method is a inductive method and it learns a predictive model to predict the structured outputs for new coming test data points.
- Brefeld and Scheffer [5] proposed a method for semi-supervised learning for structured output prediction. The method is a co-training method, and it is based on learning in a joint input-output space. It maximizes the consensus among different independent hypotheses, and extend it to a semi-supervised support vector machine learning algorithm in the joint input-output space. Moreover, the prediction loss of structured output is measured by a arbitrary structured loss function.
- Suzuki et al. [21] proposed a semi-supervised structured output prediction method for sequence labeling task. This method is based on a combination of both generative and discriminative models. The objective of this method is constructed as a log-linear form, and the objective is a combination of discriminative structured predictor and generative model to use the input data points without structured output (unlabeled data points). Moreover, these unlabeled data points is utilized by the generative model to increase the sum of the discriminant functions for all outputs.
- Li and Zemel [15] proposed a max-margin method for semi-supervised structured output prediction problem. This methods can use the both the discrete optimization algorithms and high order regularization based on the unlabeled data points. This method is shown to be closely relevant to the Posterior Regularization.

Manifold learning is a popular topic in semi-supervised learning problems [11,35,9,16]. It impose that if two data points are neighboring in the input space, their outputs should also be close to each other. Because the outputs of the data points are not complete, and most of the outputs of training data points are missing, it is important to infer the missing output from the available outputs by using the neighborhood relationship in the input space. Manifold learning has been a powerful regularization method in both classification and regression problems, and usually a squared ℓ_2 norm distance is used to measure how close two outputs (binary labels, or continues responses) are. However, in structured output prediction problem, the squared ℓ_2 norm distance cannot fit the structured outputs. In [2], a manifold regularization is also used. However, due to the complexity of the structured outputs, the regularization is not performed directly in the output space, but to the "parts" of the joint input-output space. A pair of "parts" is also compared using the squared ℓ_2 norm, so that the regularization term will not bring difficulty to the optimization of the problem. It is not guaranteed that regularizing the "parts" of input-output space can lead to the neighborhood smoothness in the output space. Actually, we can measure how close a pair of structured outputs are by a predefined

structured loss function. However, due to the complexity of this loss function, it is very difficult to optimize it to solve the parameter of the predictor.

1.3 Our contributions

To solve the problem mentioned above, in this paper, we propose to regularize the structured outputs directly in the structured output space. To avoid the difficulty of optimizing the structured loss function, we introduce a slack structured output for each training data point. This slack structured output presents the optimal output, and it is also treated as a variable during the learning procedure. For the labeled data points, their true structured outputs are available, we impose their slack structured outputs to be consistent with their true structured outputs. To prorogate the structured output from the labeled data to the unlabeled data, we use the manifold information to present the connections between the data points. To present the manifold information, we construct a nearest neighbor graph in the input data space, and use it to regularize the output space directly. More specifically, if the inputs of two data points are neighbors, we also hope their slack structured outputs are close to each other. We use the structured loss function to measure how the compared structured outputs are close to each other. Moreover, to learn the predictive model, we learn the model parameter to fit the model to the slack structured outputs. In this way, we impose the slack structured outputs to be consistent to both the prediction results of the predictive model, and the structured outputs of its nearest neighbors.

The predictive model is designed as a linear function of a joint input-output representation. We construct a objective function with respect to both the slack structured outputs and the predictive model parameter. In this objective function, we minimize the losses of the prediction results of the predictive model against the slack structured outputs, and the losses of the structured outputs of each pair of neighboring data points, simultaneously. The objective is optimized by an iterative algorithm, and the slack structured outputs and the predictive model parameter are updated alternately.

The contributions of this work are of two folds:

1. We solve the problem of manifold regularization in structured output space by introducing a slack structured output for each data point, both labeled and unlabeled, and comparing a pair of structured outputs of neighboring data points by the structured loss function.
2. We propose a novel iterative algorithm to solve the slack structured outputs and the predictive model parameters simultaneously. The optimization of the slack structured outputs are regularized by both the predictive model and the manifold. Moreover, we develop an efficient gradient descent-based method to update the predictive model parameter. This method is more efficient than the most popular optimization algorithm used in structured output prediction methods, cutting plane algorithm [6,8,7,1],

because it avoids the time-consuming quadratic programming problem of cutting plane algorithm.

1.4 Paper organization

The rest parts of this paper are organized as follows. In section 2 we introduce the proposed method, by first modeling the problem as a minimization problem, then solving it using an alternate optimization strategy, and finally developing an iterative algorithm. In section 3, the proposed is studied experimentally. It is compared to state-of-the-art semi-supervised structured output prediction methods. Its sensitivity to parameter and running time is also studied. In section 4, we give the conclusion and the future works.

2 Proposed method

In this section, we introduce the proposed method. The problem is modeled as a formulation of minimization problem, and it is then solved by a alternate optimization method with an iterative algorithm.

2.1 Problem modeling

We consider a problem of structured output prediction problem, where the input is a d-dimensional input vector, $\mathbf{x} \in \mathbb{R}^d$, and the output is a structured output, $y \in \mathcal{Y}$, where $\mathcal{Y}$ is the structured output space. We assume we have a training set of data points $\mathcal{X} = \{\theta_i\}_{i=1}^n$, where θ_i is the i-th data point, and n is the number of the data points in $\mathcal{X}$. $\mathcal{X}$ is composed of two subsets, $\mathcal{X} = \mathcal{L} \bigcup \mathcal{U}$, where $\mathcal{L}$ is the labeled data point set, and $\mathcal{U}$ is the unlabeled data point set. The data points of $\mathcal{L}$ are presented as a input-output pairs, $\theta_i = (\mathbf{x}_i, y_i)|_{i:\theta_i \in \mathcal{L}}$, where $\mathbf{x}_i \in \mathbb{R}^d$ input vector of the i-th data point, $y_i \in \mathcal{Y}$ is its corresponding structured output. The data points in $\mathcal{U}$ only have the inputs while the structured outputs are missing, $\theta_i = \mathbf{x}_i|_{i:\theta_i \in \mathcal{U}}$. To learn the missing structured outputs for the data points in $\mathcal{U}$, and a predictive model to predict the structured output for a test input, we consider the following problems to model the objective function.

2.1.1 Regularizing the structured outputs by manifold

We want to regularize the structured output by the manifold, but for the data points in $\mathcal{U}$, the structured outputs are missing. To solve this problem, we introduce a slack structured output, $z_i \in \mathcal{Y}$, for each data point $\theta_i|_{i:\theta_i \in \mathcal{X}}$. This slack structured output z_i presents the optimal output we want to learn for the i-th data point.

For a labeled data point, $\theta_i|_{i:\theta_i \in \mathcal{L}}$, since its true structured output y_i is known, we impost $z_i = y_i$. For these unlabeled data points, $\theta_i|_{i:\theta_i \in \mathcal{U}}$, we want

to predict their slack outputs by prorogating the output information from the labeled data points via a manifold. To present the manifold information, we construct a nearest neighbor graph from the input of data points of $\mathcal{X}$. For the input vector $\mathbf{x}_i$ of each data point θ_i from the inputs of data points in $\mathcal{X}$, we find its K nearest neighbors and denote the set of its nearest neighbors as $\mathcal{N}_i$. To construct the graph, we treat each data point as a node of the graph, and put a edge between the i-th node and the j-th node if $\mathbf{x}_j \in \mathcal{N}_i$. Denoting $\mathcal{E}$ as the set of edges, we have

$$\mathcal{E} = \{(\theta_i, \theta_j) : \theta_i, \theta_j \in \mathcal{X}, \mathbf{x}_j \in \mathcal{N}_i\}. \tag{1}$$

The weight of the egde (θ_i, θ_j), ω_{ij}, is assigned as a Gaussian kernel of the Euclidian distance between $\mathbf{x}_i$ and $\mathbf{x}_j$,

$$\omega_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma}\right), & if\ (\theta_i, \theta_j) \in \mathcal{E} \\ 0, & otherwise. \end{cases} \tag{2}$$

ω_{ij} is a measurement of the similarity between a pair of neighboring data points in the input space. We try to map the similarity relationship from the input space to the structured space. For a pair neighboring data points, if there are similar in the input space, i.e., ω_{ij} is large, their structured outputs should also be similar to each other, i.e., z_i and z_j are close to each other. To measure how z_i and z_j are close to each other, we use a structured loss function, $\Delta(z_i, z_j)$, to compare z_i against z_j. $\Delta(z_i, z_j)$ is a loss function to measure the loss if a structured label z_j is wrongly predicted as z_i. For example, when the structured output are the nodes of a tree, $\Delta(z_i, z_j)$ is defined as the height of the first common ancestor of z_i and z_j in the tree. Naturally, if ω_{ij} is large, we hope $\Delta(z_i, z_j)$ is as mall as possible. Thus we propose to minimize $\Delta(z_i, z_j)$ weighted by ω_{ij} with regard to z_i and z_j,

$$\begin{aligned} \min_{z_1, \cdots, z_n} & \left\{ M(z_1, \cdots, z_n) = \sum_{i,j:(\theta_i,\theta_j)\in\mathcal{E}} \omega_{ij} \Delta(z_i, z_j) \right\}, \\ & s.t.\ z_i = y_i, \forall\ i : \theta_i \in \mathcal{L}. \end{aligned} \tag{3}$$

In this way, we regularize the learning of slack structured outputs directly by the manifold, instead of regularizing the joints input-output space.

2.1.2 Learning predictive model

The problem of structured output prediction is to learn a predictive model f to predict a true structured output $y \in \mathcal{Y}$ from a input $\mathbf{w} \in \mathbb{R}^d$,

$$y \leftarrow f(\mathbf{x}; \mathbf{w}) \tag{4}$$

where $\mathbf{w}$ is parameter of the predictive model f. To design the predictive model, we present a joint representation function to match a input $\mathbf{x}$ against a candidate structured output $y' \in \mathcal{Y}$, $\Phi(\mathbf{x}, y') \in \mathbb{R}^m$, where m is the dimension

of the joint representation. An example of this representation function is for the vector output, where y' is a vector, and $\Phi(\mathbf{x}, y') = \mathbf{x} \bigotimes y'$ is the Hadamard product of $\mathbf{x}$ and y'. We further design a matching function, $g(\mathbf{x}, y'; \mathbf{w})$, to obtain the matching score of $\mathbf{x}$ and y',

$$g(\mathbf{x}, y'; \mathbf{w}) = \mathbf{w}^\top \Phi(\mathbf{x}, y') \tag{5}$$

where $\mathbf{w} \in \mathbb{R}^m$ is the parameter of the matching function. The predictive model is based on the matching function, and it returns the optimal candidate structured output, y^*, that maximized the matching scores,

$$y^* \leftarrow f(\mathbf{x}; \mathbf{w}) = \arg\max_{y' \in \mathcal{Y}} \mathbf{w}^\top \Phi(\mathbf{x}, y') \tag{6}$$

The prediction error can can be measured by a loss function, $\Delta(y^*, y)$, to compare the predicted structured output, y^*, against the true structured output, y. The problem of structured output prediction is changed to the learning of the parameter vector $\mathbf{w}$.

Since for the data points in $\mathcal{U}$, the true structured outputs are missing, we use the slack structured outputs to guide the learning of the model parameter. We hope with the learned parameter vector, $\mathbf{w}$, for the i-th training data point, the loss of predicting z_i as y_i^*, $\Delta(y_i^*, z_i)$, can be minimized. Thus we have the following optimization problem,

$$\begin{aligned} \min_{\mathbf{w}, z_1, \cdots, z_n} \quad & \sum_{i=1}^{n} \Delta(y_i^*, z_i), \\ s.t.\ & z_i = y_i, \forall\ i : \theta_i \in \mathcal{L}. \end{aligned} \tag{7}$$

where y_i^* is the predicted structured output of the i-th data point.

Due to the complexity of the loss function Δ, this problem is hard to optimize with regard to $\mathbf{w}$ directly. Instead of minimizing $\Delta(y_i^*, z_i)$ directly, we seek and minimize its upper bound. According to (6),

$$\begin{aligned} &\mathbf{w}^\top \Phi(\mathbf{x}_i, y_i^*) \geq \mathbf{w}^\top \Phi(\mathbf{x}_i, z_i), \forall\ z_i \in \mathcal{Y}, \\ &\Rightarrow \mathbf{w}^\top \left(\Phi(\mathbf{x}_i, y_i^*) - \Phi(\mathbf{x}_i, z_i)\right) + \Delta(y_i^*, z_i) \geq \Delta(y_i^*, z_i). \end{aligned} \tag{8}$$

We replace the predicted structured output y_i^* in (8) by a strutted output y_i'' to maximize the left hand of the list line of (8), so that

$$\begin{aligned} &\max_{y_i'' \in \mathcal{Y}} \left[\mathbf{w}^\top \left(\Phi(\mathbf{x}_i, y_i'') - \Phi(\mathbf{x}_i, z_i)\right) + \Delta(y_i'', z_i)\right] \\ &\geq \mathbf{w}^\top \left(\Phi(\mathbf{x}_i, y_i^*) - \Phi(\mathbf{x}_i, z_i)\right) + \Delta(y_i^*, z_i) \\ &\geq \Delta(y_i^*, z_i). \end{aligned} \tag{9}$$

Thus a upper bound of $\Delta(y_i^*, z_i)$ is obtained as follows,

$$\begin{aligned} &\max_{y_i'' \in \mathcal{Y}} \left[\mathbf{w}^\top \left(\Phi(\mathbf{x}_i, y_i'') - \Phi(\mathbf{x}_i, z_i)\right) + \Delta(y_i'', z_i)\right] \\ &= \mathbf{w}^\top \left(\Phi(\mathbf{x}_i, \upsilon_i) - \Phi(\mathbf{x}_i, z_i)\right) + \Delta(\upsilon_i, z_i), \end{aligned} \tag{10}$$

where υ_i is the structured output that maximize the left hand of (10),

$$\upsilon_i = \arg\max_{y_i'' \in \mathcal{Y}} \left[\mathbf{w}^\top \left(\Phi(\mathbf{x}_i, y_i'') - \Phi(\mathbf{x}_i, z_i)\right) + \Delta(y_i'', z_i)\right]. \tag{11}$$

Replacing $\Delta(y_i^*, z_i)$ by its upper bound in (10), we rewrite (7) as

$$\begin{aligned} \min_{\mathbf{w}, z_1, \cdots, z_n} \quad & \left\{ L(\mathbf{w}, z_1, \cdots, z_n) = \sum_{i=1}^{n} \left[\mathbf{w}^\top \left(\Phi(\mathbf{x}_i, \upsilon_i) - \Phi(\mathbf{x}_i, z_i)\right) + \Delta(\upsilon_i, z_i)\right] \right\}, \\ & s.t.\ z_i = y_i, \forall\ i : \theta_i \in \mathcal{L}. \end{aligned} \tag{12}$$

In this way, we transfer the problem of minimizing $\Delta(y_i^*, z_i)$ to the minimization of its upper bound.

2.1.3 Reducing the model complexity

To avoid the over-fitting problem, we try to reduce the complexity of the model. The complexity of the model can be measured by the squared ℓ_2 norm of the model parameter vector, $\|\mathbf{w}\|_2^2$. To reduce the complexity, we propose to minimize a regularization term $R(\mathbf{w})$,

$$\min_{\mathbf{w}} \left\{ R(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2 \right\}. \tag{13}$$

2.1.4 Overall optimization problem

The overall optimization problem of the proposed method is a combination of the three terms in (3), (12), and (23),

$$\begin{aligned} \min_{\mathbf{w}, z_1, \cdots, z_n} \quad & \Big\{ O(\mathbf{w}, z_1, \cdots, z_n) \\ & = M(z_1, \cdots, z_n) + C_1 L(\mathbf{w}, z_1, \cdots, z_n) + C_2 R(\mathbf{w}) \\ & = \sum_{i,j:(\theta_i,\theta_j)\in\mathcal{E}} \omega_{ij} \Delta(z_i, z_j) \\ & + C_1 \sum_{i=1}^{n} \left[\mathbf{w}^\top \left(\Phi(\mathbf{x}_i, \upsilon_i) - \Phi(\mathbf{x}_i, z_i)\right) + \Delta(\upsilon_i, z_i)\right] + \frac{C_2}{2}\|\mathbf{w}\|_2^2 \Big\}, \\ & s.t.\ z_i = y_i, \forall\ i : \theta_i \in \mathcal{L}, \end{aligned} \tag{14}$$

where C_1 and C_2 are the tradeoff parameters. The first term of the objective function is to regularize the slack structured outputs by the manifold, the second term is to reduce the loss of prediction error, and the last term is to reduce the complexity of the model. In this problem, the learning of the slack structured outputs are regularized by three information sources: the manifold, the known true structured outputs of the labeled data points, and the prediction results of the predictive model.

2.2 Problem optimization

To solve the problem in (14), we use an alternate optimization strategy. In an iterative algorithm, when the model parameter vector $\mathbf{w}$ is considered, the slack structured outputs $z_1, \cdots, z_n$ are fixed. When $z_1, \cdots, z_n$ are considered, $\mathbf{w}$ is considered. In the following subsections, we will discuss how to solve $\mathbf{w}$ and $z_1, \cdots, z_n$ respectively.

2.2.1 Solving $\boldsymbol{w}$ while fixing $z_1, \cdots, z_n$

When we consider the model parameter vector $\mathbf{w}$, the slack structured outputs $z_1, \cdots, z_n$ are fixing. We remove the terms in (14) irrelevant to $\mathbf{w}$, and obtain the following problem,

$$\min_{\mathbf{w}} \left\{ O_1(\mathbf{w}) = C_1 \sum_{i=1}^{n} \left[\mathbf{w}^\top \left(\Phi(\mathbf{x}_i, \upsilon_i) - \Phi(\mathbf{x}_i, z_i) \right) + \Delta(\upsilon_i, z_i) \right] + \frac{C_2}{2} \|\mathbf{w}\|_2^2 \right\}. \tag{15}$$

Please note that υ_i is also a function of $\mathbf{w}$ according to (11). However, because it is coupled with a maximization problem, thus it is hard to optimize with regard to $\mathbf{w}$ directly. Thus we use the strategy of expectation-maximization algorithm, update υ_i by using the solutions of $\mathbf{w}$ and $z_1, \cdots, z_n$ in previous iteration, and then fix it when $\mathbf{w}$ is optimized in current iteration. After υ_i is fixed, we use the gradient descent algorithm to update $\mathbf{w}$. To seek the minimization of $O_1(\mathbf{w})$, $\mathbf{w}$ should descent to the direction of gradient. The gradient function of $O_1(\mathbf{w})$ is

$$\nabla O_1(\mathbf{w}) = C_1 \sum_{i=1}^{n} \left(\Phi(\mathbf{x}_i, \upsilon_i) - \Phi(\mathbf{x}_i, z_i) \right) + C_2 \mathbf{w}. \tag{16}$$

The updating rule is

$$\begin{aligned} \mathbf{w} \leftarrow & \mathbf{w} - \eta \nabla O_1(\mathbf{w}) \\ = & \mathbf{w} - \eta \left[C_1 \sum_{i=1}^{n} \left(\Phi(\mathbf{x}_i, \upsilon_i) - \Phi(\mathbf{x}_i, z_i) \right) + C_2 \mathbf{w} \right] \\ = & (1 - \eta C_2)\mathbf{w} - \eta C_1 \sum_{i=1}^{n} \left(\Phi(\mathbf{x}_i, \upsilon_i) - \Phi(\mathbf{x}_i, z_i) \right), \end{aligned} \tag{17}$$

where η is the descent step.

2.2.2 Solving $z_1, \cdots, z_n$ while fixing $\boldsymbol{w}$

We fix the $\mathbf{w}$ when $z_1, \cdots, z_n$ are considered, and remove the terms irrelevant to $z_1, \cdots, z_n$. The following problem is obtained,

$$\begin{aligned} \min_{z_1,\cdots,z_n} \quad & \Bigg\{ O_2(z_1, \cdots, z_n) = \sum_{i,j:(\theta_i,\theta_j)\in\mathcal{E}} \omega_{ij} \Delta(z_i, z_j) \\ & + C_1 \sum_{i=1}^{n} \left[-\mathbf{w}^\top \Phi(\mathbf{x}_i, z_i) + \Delta(\upsilon_i, z_i) \right] \Bigg\}, \\ s.t.\ & z_i = y_i, \forall\ i : \theta_i \in \mathcal{L}. \end{aligned} \tag{18}$$

It is difficult to optimize all the slack structured outputs $z_1, \cdots, z_n$ simultaneously. Thus we chose to update them one by one. When one slack structured output z_i is considered, other ones $z_j|_{j\neq i}$ are fixed. In this case, we obtain the following problem for the i-th data point,

$$\begin{aligned} \min_{z_i} \quad & \Bigg\{ O_3(z_i) = \sum_{j:(\theta_i,\theta_j)\in\mathcal{E}} \omega_{ij} \Delta(z_i, z_j) + \sum_{j':(\theta_{j'},\theta_i)\in\mathcal{E}} \omega_{j'i} \Delta(z_{j'}, z_i) \\ & + C_1 \left[-\mathbf{w}^\top \Phi(\mathbf{x}_i, z_i) + \Delta(\upsilon_i, z_i) \right] \Bigg\}, \\ s.t.\ & z_i = y_i, \forall\ i : \theta_i \in \mathcal{L}. \end{aligned} \tag{19}$$

From the formulation, we can see that the optimal z_i should be consistent to the slack structured outputs of its nearest neighbors, and the prediction result of the predictive model. The solution for this problem can be obtained by a linear search in the structured output space,

$$z_i = \begin{cases} \arg\max_{y'\in\mathcal{Y}} O_3(y_i'), & if\ \theta_i \in \mathcal{U} \\ y_i, & otherwise. \end{cases} \tag{20}$$

2.3 Iterative algorithm

We summarize the developed iterative learning algorithm in Algorithm (1). From this algorithm, we can see that the iterations are repeated T times. In each iteration, we first update υ_i and z_i for each data point, and then update $\mathbf{w}$. This algorithm is named as manifold regularized structured output learning algorithm (MRSO).

Algorithm 1 Iterative algorithm of MRSO.

Input: Training set of data points $\mathcal{X}$;
Input: Tradeoff parameters C_1 and C_2;
Input: Maximum number of iterations, T;
Initialize model parameter vector $\mathbf{w}^0$;
Initialize the slack structured outputs $z_1^0, \cdots, z_n^0$;
for $t = 1, \cdots, T$ **do**
  for $i = 1, \cdots, n$ **do**
    Update υ_i^t of the i-th data point by fixing z_i^{t-1} and $\mathbf{w}^{t-1}$,

$$\upsilon_i^t = \arg\max_{y_i'' \in \mathcal{Y}} \left[{\mathbf{w}^{t-1}}^\top \left(\Phi(\mathbf{x}_i, y_i'') - \Phi(\mathbf{x}_i, z_i^{t-1}) \right) + \Delta(y_i'', z_i^{t-1}) \right]. \tag{21}$$

    Update z_i^t of the i-th data point by fixing $\mathbf{w}^{t-1}$, $z_j^{t-1}|_{j \neq i}$ and υ_i^t.
    if $\theta_i \in \mathcal{U}$ **then**

$$\begin{aligned} z_i^t = & \arg\min_{y_i' \in \mathcal{Y}} \left\{ \sum_{j:(\theta_i, \theta_j) \in \mathcal{E}} \omega_{ij} \Delta(y_i', z_j^{t-1}) + \sum_{j':(\theta_{j'}, \theta_i) \in \mathcal{E}} \omega_{j'i} \Delta(z_{j'}^{t-1}, y_i') \right. \\ & \left. + C_1 \left[-{\mathbf{w}^{t-1}}^\top \Phi(\mathbf{x}_i, y_i') + \Delta(\upsilon_i^t, y_i') \right] \right\}; \end{aligned} \tag{22}$$

    else
      $z_i^t = y_i$;
    end if
  end for
  Update $\mathbf{w}^t$ by fixing $\upsilon_1^t, \cdots, \upsilon_n^t$, and $z_1^t, \cdots, z_n^t$,

$$\mathbf{w}^t = (1 - \eta C_2)\mathbf{w}^{t-1} - \eta C_1 \sum_{i=1}^{n} \left(\Phi(\mathbf{x}_i, \upsilon_i^t) - \Phi(\mathbf{x}_i, z_i^t) \right); \tag{23}$$

end for
Output: $\mathbf{w}^T$ and $z_1^T, \cdots, z_n^T$.

3 Experiments

3.1 Data sets

– The first data set we used is Cora data set [19]. The output of this data set is the class label vector of multi-class classification problem. This data set is a linked computer science paper data set. Each paper is a treated as a data point. In this data set, there are 9,947 data points. The papers without a reference list is removed from the data set, and 9,555 papers are left. All the papers belong to the 8 classes. To construct a feature vector from a paper, we extract a term frequency vector, and a link view vector, and concatenate them as a feature vector [38].
  For each data point, $\mathbf{x}_i$, we construct a vector output, $\mathbf{y}_i = [y_{i1}, \cdots, y_{i8}] \in \{1, 0\}^8$, as the structured output. This vector, $\mathbf{y}_i$, is a 8-dimensional binary

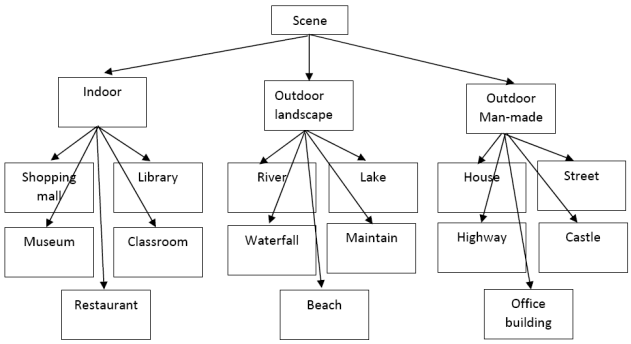


Fig. 1 Tree structured outputs of SUN data set.

vector. If this data point belongs to the k-class, then the k-th element of this vector is 1, or 0 otherwise,

$$y_{ik} = \begin{cases} 1, \, if \ \mathbf{x}_i \ belongs \ to \ the \ k-th \ class, \\ 0, \qquad\qquad otherwise. \end{cases} \tag{24}$$

We further define the joint input-output representation function as $\Phi(\mathbf{x}, \mathbf{y}) = \mathbf{x} \otimes \mathbf{y}$. To measure the prediction error of predicting $\mathbf{y}_i$ as $\mathbf{y}_i^*$ by the $0-1$ loss, and define $\Delta(y_i^*, y_i)$,

$$\Delta(\mathbf{y}_i^*, \mathbf{y}_i) = \begin{cases} 1, \, if \ \mathbf{y}_i^* = \mathbf{y}_i, \\ 0, \, otherwise. \end{cases} \tag{25}$$

– The second data set is SUN data set [34]. The outputs of this data set are the nodes of a tree structure. In this data set, there a 2,000 images, and they belongs to 15 different classes of scenes. The classes are organized as a scene tree. The root node is scene, and it has three child nodes, which are indoor, outdoor land space, and outdoor man-made. These three child nodes have further 15 leaf nodes, which are the 15 classes. Thus there are 19 nodes in the tree in total. The scene tree is shown in figure 1. Each image belongs to one of the classes. To represent the image, we extract the HOG features from the image and use them as visual features. In this case, the structured output is a node of the tree. We present a output of the i-th data point by using a 19-dimensional binary vector $\mathbf{y}_i \in \{1, 0\}^{19}$. The k-th element of $\mathbf{y}_i$ is defined as

$$y_{ik} = \begin{cases} 1, \ \ if \ the \ k-th \ node \ is \ the \ class \ of \ \mathbf{x}_i, \\ \quad or \ it \ is \ a \ ancestor \ of \ the \ class \ of \ \mathbf{x}_i, \\ 0, \qquad\qquad\qquad otherwise. \end{cases} \tag{26}$$

We also define the joint input-output representation function as $\Phi(\mathbf{x}, \mathbf{y}) = \mathbf{x} \otimes \mathbf{y}$. The structured loss function $\Delta(\mathbf{y}_i^*, \mathbf{y}_i)$ is defined as the height of the first common ancestor of the predicted output $\mathbf{y}_i^*$ and true output $\mathbf{y}_i$.

– The third data set is a subset of Biocreative data set, provided by the special session of CoNLL2002 [23]. The outputs of this data set is label sequences. This set contains 500 sentences from biomedical papers. Each word in a sentences can be labeled as one of the nine named entities. The problem is to assign a sequence of named entity labels to a sentence. Thus the output of a sentence of m words, $\mathbf{x}_i$, is a sequence of labels, $y_i = (y_{i1}, \cdots, y_{im})$, where y_{ik} is the label of the k-th word. The joint input-output representation function, $\Phi(\mathbf{x}_i, y_i)$ is defined as the histogram of state transition and a set of features describing the emissions [22]. The structured loss function to compare a predicted label sequence y_i^* against the truce label sequence y_i is defined as follows,

$$\Delta(y_i^*, y_i) = \begin{cases} 1, \, if \; y_i^* = y_i, \\ 0, \, otherwise. \end{cases} \tag{27}$$

3.2 Experiment setup

To perform the experiment, we employ the 10-fold cross validation. A entire data set is split into ten folds randomly. Each fold is used as a test set in turn. The rest nine folds are combined as a training set. Moreover, we further select two folds from the training set randomly as labeled data set, and leave the rest seven folds as unlabeled data set. The proposed method is applied to the training set to learn the predictive model parameter, and the structured outputs of the unlabeled training data points. Moreover, the learned predictive model are also applied to the test set to predict the structured outputs of the test data points. The prediction performance is evaluated by the average structured loss (ASL) over the test set, $\mathcal{T}$,

$$ASL = \frac{1}{|\mathcal{T}|} \sum_{i:\theta_i \in \mathcal{T}} \Delta(y_i^*, y_i). \tag{28}$$

3.3 Experiment results

In this section, we study the proposed method experimentally. We first compare it to the state-of-the-art semi-supervised structured output prediction methods. Then we study the convergency of the proposed iterative algorithm. Finally, we study how the algorithm performs over different tradeoff parameters.

3.3.1 Comparison to state-of-the-art

We compare the proposed MRSO algorithm against several state-of-the-art semi-supervised learning methods for structured output prediction. We list them as follows:

- Semi-supervised structured (STR) max-margin optimization method [2],
- Co-support vector learning for structured output variables (CoSVM) [5],
- Semi-supervised structured output learning based on a hybrid generative and discriminative models (HySOL) [21], and
- High order regularization for semi-supervised learning of structured output problems (HOR) [15].

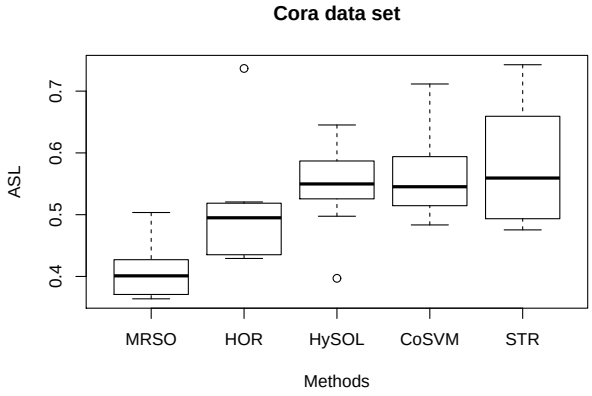
The boxplots of the 10-fold cross validation are given in figure 2. From results in figure 2, we can easily determine that the proposed MRSO algorithm outperforms the other algorithms over all three data sets. For example, in figure 2(a), we can see that the median value of the ASL values of the MRSO is as low as about 0.4, while the median ASL of the second best method, HOR, is as high as 0.5. For all other three methods, the media values of ASL are higher than 0.5, which are around 0.55. The outperforming of the proposed algorithm MRSO over the compared methods is even more obvious in 2(b). In this figure, only the proposed MRSO method achieves a median value of ASL lower than 0.6, and those median values of the compared methods are higher than 0.7. Moreover, it seems that HOR and HySOL performs better than CoSVM and STR.

3.3.2 Algorithm convergency

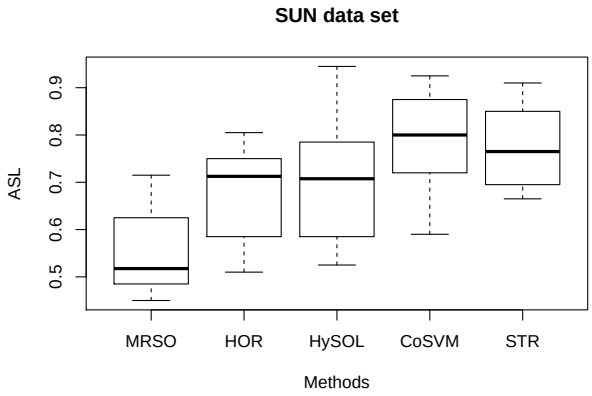
The proposed algorithm is an iterative algorithm. We also study the convergency of the algorithm by plotting the responses of the objective function of different iterations. This experiment is conducted over the Cora data set. The curve is given in figure 3. From this figure, we can observe the iterative algorithm can converge at some point of iteration. For example, the objective decreases significantly from the first iteration to the 60-th iteration, and then the objective stays stable after the 60-the iteration. This indicates the algorithm converges.

3.3.3 Tradeoff parameter analysis

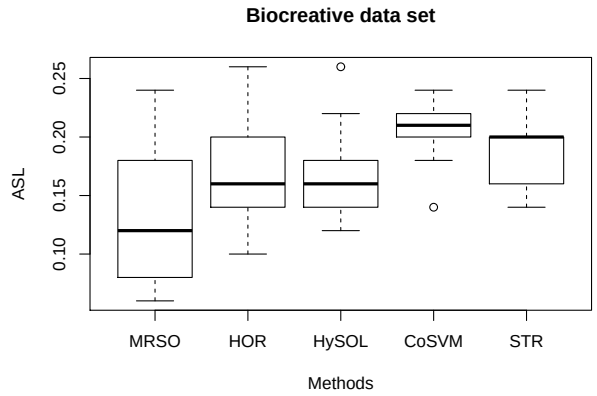
In the objective function of our formulation (14), there are two tradeoff parameters, C_1 and C_2. We also want to know how these parameters effect the performance of our algorithm. To this end, we plot the curve of the different values of ASL of different values of C_1 and C_2. The curves are shown in figure 4. Please note that the data in figure 4 is obtained by conducting experiments in Cora data set. From this figure, we can observe that our algorithm is table to both the parameters. In figure 4(a), when the parameter C_1 varies from 0.1 to 1000, the range of ALS of MRSO is $[0.40, 0.45]$, and the variance is very small. Moreover, in figure 4(b), we can also observe that the range of ALS of MRSO is $[0.40, 0.43]$ when C_2 is varied.



(a) Core data set



(b) SUN data set



(c) Biocreative data set

Fig. 2 Results of comparison to state-of-the-art.

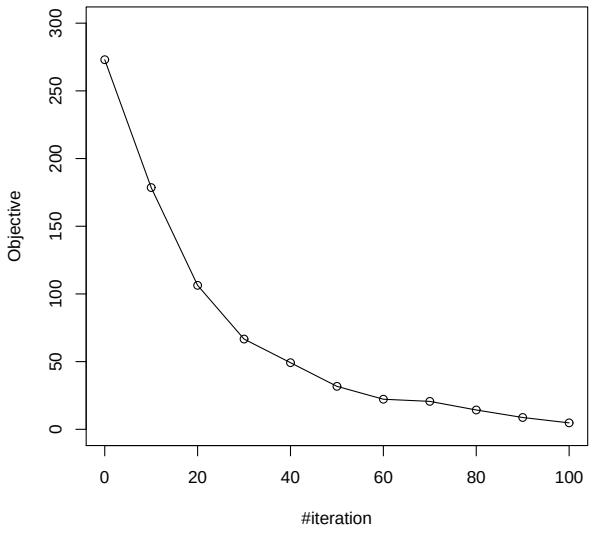


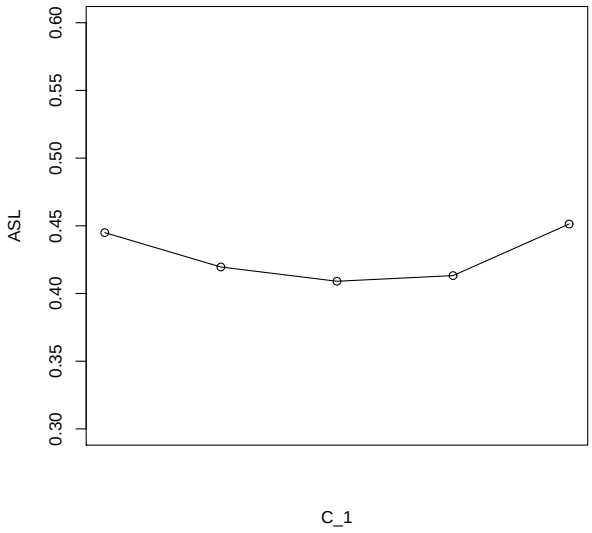
Fig. 3 Responses of objective function of different iterations.

4 Conclusion and future works

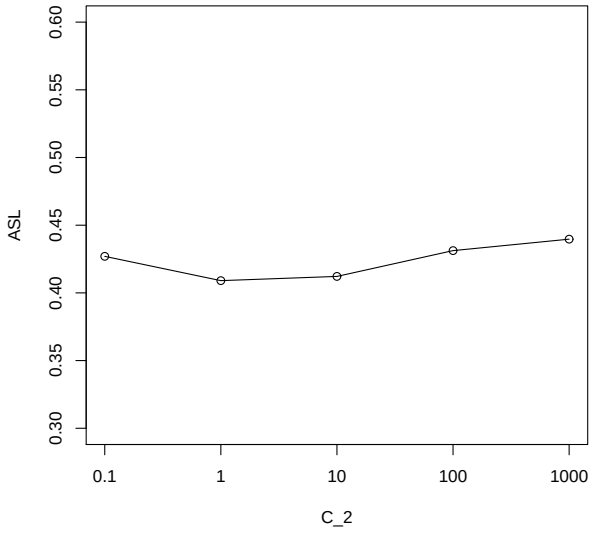
This paper investigate the problem of semi-supervised structured output prediction. We propose to use the manifold structure to regularize the structured outputs directly. However, in this problem, many training data points only have input feature vectors, while the structured outputs are missing. To solve this problem, we propose a slack structured output for each training data point, either labeled or unlabeled. Moreover, we construct a nearest neighbor graph in the input space to present the manifold structure, and use it to regularize the learning of the slack structured outputs. We impose the slack structured outputs to be consistent to both the manifold structure and the prediction results of a structured output predictor. More specifically, we use a structured loss function to measure how a pair of structured output fits to the manifold distribution. A unified objective is constructed for the learning of both slack structured outputs and the predictive model parameter, and an iterative algorithm is proposed to minimize this objective function. The experiment results show that the proposed algorithm outperforms the state-of-the-art semi-supervised structured output prediction methods.

References

1. Abdelouadoud, S., Girard, R., Neirac, F., Guiot, T.: Optimal power flow of a distribution system based on increasingly tight cutting planes added to a second order cone



(a) C_1



(b) C_2

Fig. 4 Sensitivity curve of tradeoff parameters.

relaxation. International Journal of Electrical Power and Energy Systems **69**, 9–17 (2015)
2. Altun, Y., McAllester, D., Belkin, M.: Maximum margin semi-supervised learning for structured variables. In: Advances in Neural Information Processing Systems, pp. 33–40 (2005)
3. Astikainen, K., Holm, L., Pitkänen, E., Szedmak, S., Rousu, J.: Structured output prediction of novel enzyme function with reaction kernels. Communications in Computer and Information Science **127 CCIS**, 367–379 (2011)
4. Braida, F., Mello, C.E., Pasinato, M.B., Zimbrao, G.: Transforming collaborative filtering into supervised learning. Expert Systems with Applications **42**(10), 4733–4742 (2015)
5. Brefeld, U., Scheffer, T.: Semi-supervised learning for structured output variables. In: ICML 2006 - Proceedings of the 23rd International Conference on Machine Learning, vol. 2006, pp. 145–152 (2006)
6. Chouman, M., Crainic, T.: Cutting-plane matheuristic for service network design with design-balanced requirements. Transportation Science **49**(1), 99–113 (2015)
7. Eronen, V.P., Mäkelä, M., Westerlund, T.: Extended cutting plane method for a class of nonsmooth nonconvex minlp problems. Optimization **64**(3), 641–661 (2015)
8. Fang, Y., Chu, F., Mammar, S., Shi, Q.: A new cut-and-solve and cutting plane combined approach for the capacitated lane reservation problem. Computers and Industrial Engineering **80**, 212–221 (2015)
9. Feng, J., Wang, J., Zhang, H., Han, Z.: Fault diagnosis method of joint fisher discriminant analysis based on the local and global manifold learning and its kernel version. IEEE Transactions on Automation Science and Engineering (2015). DOI 10.1109/TASE.2015.2417882
10. Han, Y., Wei, X., Cao, X., Yang, Y., Zhou, X.: Augmenting image descriptions using structured prediction output. IEEE Transactions on Multimedia **16**(6), 1665–1676 (2014)
11. Ho, S., Dai, P., Rudzicz, F.: Manifold learning for multivariate variable-length sequences with an application to similarity search. IEEE Transactions on Neural Networks and Learning Systems (2015). DOI 10.1109/TNNLS.2015.2399102
12. Joachims, T.: Structured output prediction with support vector machines. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **4109 LNCS**, 1–7 (2006)
13. Kajdanowicz, T., Wozniak, M., Kazienko, P.: Multiple classifier method for structured output prediction based on error correcting output codes. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **6592 LNAI**(PART 2), 333–342 (2011)
14. Kim, M., Pavlovic, V.: Structured output ordinal regression for dynamic facial emotion intensity prediction. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **6313 LNCS**(PART 3), 649–662 (2010)
15. Li, Y., Zemel, R.: High order regularization for semi-supervised learning of structured output problems. In: 31st International Conference on Machine Learning, ICML 2014, vol. 4, pp. 3205–3217 (2014)
16. Lorente, D., Escandell-Montero, P., Cubero, S., Gómez-Sanchis, J., Blasco, J.: Visible-nir reflectance spectroscopy and manifold learning methods applied to the detection of fungal infections on citrus fruit. Journal of Food Engineering **163**, 17–24 (2015)
17. Luo, J., Brodsky, A.: An em-based multi-step piecewise surface regression learning algorithm. In: The seventh international conference on data mining (WORLDCOMP DMIN 11), pp. 286–292 (2011)
18. Oonk, S., Spijker, J.: A supervised machine-learning approach towards geochemical predictive modelling in archaeology. Journal of Archaeological Science **59**, 80–88 (2015)
19. Sen, P., Namata, G.M., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T.: Collective classification in network data. AI Magazine **29**(3), 93–106 (2008)
20. Su, H., Heinonen, M., Rousu, J.: Structured output prediction of anti-cancer drug activity. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **6282 LNBI**, 38–49 (2010)

21. Suzuki, J., Fujino, A., Isozaki, H.: Semi-supervised structured output learning based on a hybrid generative and discriminative approach. In: EMNLP-CoNLL 2007 - Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 791–800 (2007)
22. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: Proceedings of the twenty-first international conference on Machine learning, p. 104. ACM (2004)
23. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. In: Journal of Machine Learning Research, pp. 1453–1484 (2005)
24. Wang, H., Wang, J.: An effective image representation method using kernel classification. In: Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on, pp. 853–858. IEEE (2014)
25. Wang, J., Wan, J., Liu, Z., Wang, P.: Data mining of mass storage based on cloud computing. In: Grid and Cooperative Computing (GCC), 2010 9th International Conference on, pp. 426–431. IEEE (2010)
26. Wang, J., Wang, H., Zhou, Y., McDonald, N.: Multiple kernel multivariate performance learning using cutting plane algorithm. In: Systems, Man and Cybernetics (SMC), 2015 IEEE International Conference on. IEEE (2015)
27. Wang, J., Zhou, Y., Yin, M., Chen, S., Edwards, B.: Representing data by sparse combination of contextual data points for classification. In: Advances in Neural Networks–ISNN 2015. Springer (2015)
28. Wang, K., Zhou, X., Chen, H., Lang, M., Raicu, I.: Next generation job management systems for extreme-scale ensemble computing. In: Proceedings of the 23rd international symposium on High-performance parallel and distributed computing, pp. 111–114. ACM (2014)
29. Wang, K., Zhou, X., Li, T., Zhao, D., Lang, M., Raicu, I.: Optimizing load balancing and data-locality with data-aware scheduling. In: Big Data (Big Data), 2014 IEEE International Conference on, pp. 119–128. IEEE (2014)
30. Wang, K., Zhou, X., Qiao, K., Lang, M., McClelland, B., Raicu, I.: Towards scalable distributed workload manager with monitoring-based weakly consistent resource stealing. In: Proceedings of the 24rd international symposium on High-performance parallel and distributed computing, pp. 219–222. ACM (2015)
31. Wang, Y., Yang, T., Ma, Y., Halade, G.V., Zhang, J., Lindsey, M.L., Jin, Y.F.: Mathematical modeling and stability analysis of macrophage activation in left ventricular remodeling post-myocardial infarction. BMC genomics **13**(Suppl 6), S21 (2012)
32. Wu, Y., Yuan, Z., Liu, Y., Zheng, N.: Discriminative structured outputs prediction model and its efficient online learning algorithm. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops 2009, pp. 2087–2094 (2009)
33. Xia, P., Liu, B., Sun, Y., Chen, C.: Reciprocal recommendation system for online dating. In: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. ACM (2015)
34. Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 3485–3492 (2010). DOI 10.1109/CVPR.2010.5539970
35. Xing, X., Wang, K., Lv, Z., Zhou, Y., Du, S.: Fusion of local manifold learning methods. IEEE Signal Processing Letters **22**(4), 395–399 (2015)
36. Xu, L., Zhan, Z., Xu, S., Ye, K.: Cross-layer detection of malicious websites. In: Proceedings of the third ACM conference on Data and application security and privacy, pp. 141–152. ACM (2013)
37. Xu, L., Zhan, Z., Xu, S., Ye, K.: An evasion and Counter-Evasion study in malicious websites detection. In: 2014 IEEE Conference on Communications and Network Security (CNS) (IEEE CNS 2014). San Francisco, USA (2014)
38. Zhang, H., Jiao, Y., Zhang, Y., Shimada, K.: Automated segmentation of cerebral aneurysms based on conditional random field and gentle adaboost. Mesh Processing in Medical Image Analysis 2012 pp. 59–69 (2012)

39. Zhang, S., Caragea, D., Ou, X.: An empirical study on using the national vulnerability database to predict software vulnerabilities. In: Database and Expert Systems Applications, pp. 217–231. Springer Berlin Heidelberg (2011)